RETHINKING AGENTIC CONTINUAL LEARNING AS BAYESIAN DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly used as agents, yet they rarely turn test-time experience into persistent improvements. We connect agentic continual learning (CL) to Bayesian discovery: both accumulate evidence from sequential interactions into persistent knowledge. With the model frozen, what an agent learns can persist only as text, and we represent it as falsifiable hypotheses about the entities of its environment, whose beliefs each task updates by Bayes' rule. This view clarifies what a task stream must provide: a refutation corrects rather than interferes, forgetting arises when memory revision discards evidence, and a stream becomes silent when later tasks provide no evidence for or against what was learned earlier. We propose DISCOVERCL, a training-free harness that approximates these updates with a research loop of proposing hypotheses, acting, and recording evidence. We further introduce DISCOVERSCORE, which measures how much a stream updates an agent's beliefs per task. With Claude Sonnet 5 and Opus 5 on six CL-BENCH streams, DISCOVERCL matches or improves over the no-memory baseline in all 12 settings and is the strongest bounded-memory method in 9 of 12. With Opus 5, it recovers 50–97% of full-context ICL's gain on Database, Spectrum, Sales, and Codebase while using 2.6–9.6$\times$ less memory. DISCOVERSCORE ranges from 0.37 to 3.48 bits per task, showing that the streams differ in how much they allow the agent to learn continually.

1 INTRODUCTION

Continual learning for language agents has become an increasingly active topic (Shinn et al., 2023; Wang et al., 2024a; Zhao et al., 2024; Suzgun et al., 2026; Ouyang et al., 2026; Zheng et al., 2026; Shu et al., 2026; Asawa et al., 2026). As large language models (LLMs) are deployed as agents for reasoning, tool use, web navigation, and interaction with complex environments (Yao et al., 2022; Zhou et al., 2024; Liu et al., 2024; Xie et al., 2024), they repeatedly encounter useful facts, strategies, and workflows, yet most remain fixed after deployment. Agentic continual learning (CL) aims to turn this experience into better behavior on later tasks without updating the model's weights (Zhao et al., 2024; Suzgun et al., 2026; Ouyang et al., 2026; Shu et al., 2026). Despite this growing interest, it remains unclear what such an agent actually learns. We ask: *what should an agent learn from a task stream, and when does a stream allow it to learn?*

Existing methods answer the first question implicitly through memory. They design diverse memory systems that store trajectories, skills, or workflows, and revise and retrieve them for later tasks (Shinn et al., 2023; Wang et al., 2024a; Zhao et al., 2024; Wang et al., 2025; Suzgun et al., 2026; Ouyang et al., 2026). Benchmarks then run such agents over a fixed set of pre-designed tasks in one prescribed order and report whether downstream performance improves (Shu et al., 2026; Asawa et al., 2026). This practice has two problems: (**1**) a single fixed-order gain entangles the memory method with the learning opportunities supplied by that particular curriculum, without revealing which later tasks test earlier knowledge or where the stream becomes silent; and (**2**) memory is revised without evidence, since stored items are rewritten, merged, or pruned freely, so a claim can disappear without ever being refuted. The final gain therefore cannot identify what the agent learned or which parts of the stream enabled that learning.

Key challenges. Defining it is difficult for two reasons. (**1**) *Heterogeneous tasks.* Tasks in a stream differ in structure, action space, and harness, so there is no common representation for what a task

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

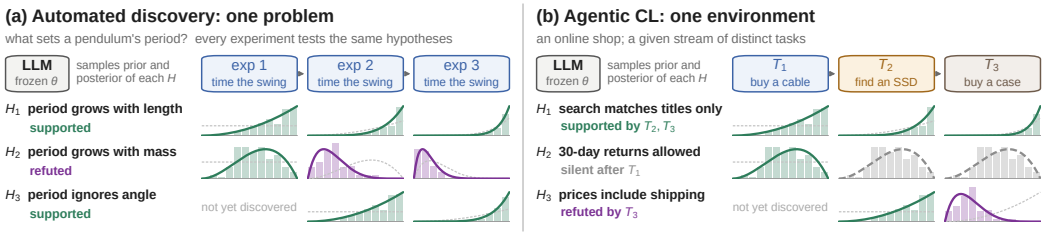

Figure 1: **Comparison between automated discovery and agentic continual learning.** (**a**) Automated discovery: every experiment is designed to test hypotheses about one problem, so each experiment moves the LLM-sampled beliefs. (**b**) Agentic continual learning: hypotheses about one environment are tested by a given stream of tasks, and a later task may be silent about them.

teaches or for accumulating it across tasks (Park et al., 2023; Wang et al., 2025; Hu et al., 2026). (**2**) *Policy-dependent data.* In conventional CL, the data of each task are fixed (Parisi et al., 2019; De Lange et al., 2022; Wang et al., 2024b); in agentic CL, the agent generates them on-policy, and what a task reveals depends on the actions it takes (Yao et al., 2022; Wang et al., 2024a; Xie et al., 2024). The same stream can be informative for one agent and uninformative for another, so learnability belongs to the stream and the agent together and cannot be read off the trajectories.

Our solution. We view agentic CL as Bayesian *discovery*: each task is an experiment that turns the agent's prior belief in a set of falsifiable hypotheses into a posterior, which becomes the prior for the next task. This answers both parts of our question. What an agent should learn is this posterior, and a stream allows learning exactly when later tasks move beliefs in hypotheses formed earlier; when they do not, the stream is *silent*. The view resolves both challenges: for (**1**), every task, whatever its structure, acts only by turning a prior into a posterior, which gives one representation for all tasks; for (**2**), learning is measured on the change in belief rather than on trajectories, so it is defined for each agent on its own evidence. It also makes memory revision accountable: a hypothesis is dropped only when refuted, so forgetting arises only when revision discards evidence (Figure 1).

Our contribution. (**1**) **A discovery view** of agentic CL that answers both parts of our question: an agent should learn posterior beliefs over hypotheses, and a stream allows learning when later tasks move these beliefs rather than stay silent (Section 4). (**2**) **DISCOVERCL**, a meta-harness that learns what the view prescribes through a three-stage loop with two additional model calls per task, while retiring a hypothesis only when refuted (Section 5). (**3**) **DISCOVERSCORE**, a meta-evaluation that measures when a stream allows a given agent to learn, as how far it moves the agent's beliefs per task (Section 6). On the six streams of CL-BENCH (Asawa et al., 2026) with Claude Sonnet 5 and Opus 5, DISCOVERCL matches or beats the no-memory baseline in all 12 settings and is the strongest bounded-memory method in 9 of 12. With Opus 5, it recovers 50–97% of full-context ICL's gain on Database, Spectrum, Sales, and Codebase while using 2.6–9.6× less memory; DISCOVERSCORE ranges from 0.37 to 3.48 bits per task, showing that streams differ widely in how much they allow an agent to learn.

2 RELATED WORKS

Continual learning for language agents. Most continual learning methods for language agents keep the model fixed and store experience in an external memory: REFLEXION (Shinn et al., 2023) records verbal feedback, EXPEL (Zhao et al., 2024) extracts reusable insights, VOYAGER (Wang et al., 2024a), Agent Workflow Memory (Wang et al., 2025), and MEMP (Fang et al., 2025b) store skills or procedures, and other methods update and consolidate reusable context as experience accumulates (Suzgun et al., 2026; Zhang et al., 2026; Ouyang et al., 2026; Xu et al., 2026; Fang et al., 2025a). DISCOVERCL shares this setting: the model is frozen and learning happens only through what is carried in context. It differs in what is carried and how it is revised. Existing memories are procedures valued by their utility and are rewritten or merged freely, so the evidence behind them is lost; DISCOVERCL carries falsifiable hypotheses with their evidence, revises them only by evidence, and drops a hypothesis only when a later task refutes it.

Benchmarks for agentic continual learning. Benchmarks order examples, feedback, or episodes into streams (Wu et al., 2024; Ai et al., 2025; Wei et al., 2025; Xi et al., 2024; Shridhar et al., 2021;

Wang et al., 2022) or build explicit cross-task structure (Zheng et al., 2025; Joshi et al., 2025; Shu et al., 2026; Asawa et al., 2026; Yang et al., 2026), and report downstream performance. We evaluate on these streams and report the same metric. DISCOVERSCORE complements it: downstream performance mixes the quality of a method with the evidence in the stream, whereas DISCOVERSCORE measures the latter directly, as how far a stream moves a given agent's beliefs, and so tells whether a stream allows that agent to learn at all.

Automated discovery and meta-harnesses. Automated-discovery systems run a task-agnostic loop that proposes, tests, and revises candidates: FUNSEARCH (Romera-Paredes et al., 2024) evaluates generated programs, the AI Scientist (Lu et al., 2024; Yamada et al., 2025) runs machine-learning experiments, AI co-scientist (Gottweis et al., 2025) ranks research hypotheses, and TTT-DISCOVER (Yuksekgonul et al., 2026) trains models on evaluator feedback; SCIENCEAGENTBENCH (Chen et al., 2025) and DISCOVERYWORLD (Jansen et al., 2024) evaluate such agents. DISCOVERCL adopts the same loop as a meta-harness that stays fixed across tasks, as agent scaffolds such as ReAct (Yao et al., 2022) do within one task. The difference lies in the experiments. Discovery systems test candidates by repeated trials on one fixed problem with a given evaluator; in DISCOVERCL, the experiments are the tasks of a stream, and the evidence each yields depends on the agent's own policy rather than on an external evaluator.

3 PRELIMINARIES

Language agents. Given a task T that specifies an instruction or goal, a language agent interacts with an environment. At step t, it receives an observation o_t and chooses an action $a_t \sim \pi_\theta(\cdot \mid T, h_t)$, where $h_t = (o_0, a_0, \ldots, a_{t-1}, o_t)$ is the interaction history and θ are the frozen model weights. A rollout ends after n steps and produces a trajectory $\tau = (T, o_0, a_0, \ldots, o_n, y)$, where y is the final outcome or evaluation signal. We write $\tau \sim \pi_\theta(T)$ for a trajectory obtained by rolling out the policy on T.

Agentic task streams. A task stream is an ordered sequence of K tasks $T_1, \ldots, T_K$ in a shared environment. Rolling out the policy on each task gives $\tau_k \sim \pi_\theta(T_k)$, and the trajectories of the first k tasks form $\mathcal{D}_k = \{\tau_1, \ldots, \tau_k\}$. Tasks may share facts, rules, procedures, or environment state, but the agent is not told which information will transfer. For example, one task may ask the agent to move an apple from a refrigerator to a table. Its trajectory reveals that the refrigerator must be opened before the apple can be taken and leaves the apple on the table; a later task that asks the agent to place the apple in a bowl can use both pieces of information.

4 RETHINKING AGENTIC CL AS BAYESIAN DISCOVERY

Agentic CL as a general form of automated research. Automated research discovers knowledge through an iterative loop of proposing hypotheses, running experiments, and updating beliefs (Lu et al., 2024; Gottweis et al., 2025; Agarwal et al., 2025). Agentic CL follows this exact loop, with the key distinction that each experiment is a task drawn from an uncontrolled stream $(T_1, \ldots, T_K)$. While traditional automated research is a special case where every task addresses the same predictable problem ($T_1 = \cdots = T_K$), an agentic CL system cannot anticipate which hypotheses the next task will evaluate. Consequently, it must maintain a hypothesis set broad enough that any incoming task tests a subset of its beliefs. We now define the ideal learner for this setting, introduce DISCOVERCL (Section 5) to approximate it, and propose DISCOVERSCORE (Section 6) to quantify how much the agent learns from a given stream.

Prior over a hypothesis set. Ideally, the agent would estimate $\boldsymbol{\phi}$, the probability that each possible claim about its environment holds, but this space of claims is open-ended and intractable. An LLM, however, can sample candidate claims, so we restrict $\boldsymbol{\phi}$ to a finite set of active hypotheses $\mathcal{H}_k$ held when task T_k arrives. Each hypothesis is a falsifiable text claim, such as H: "prices in `table1` are in dollars", which a single price in euros refutes. In the ideal case the hypotheses are independent, so, following Agarwal et al. (2025), we model the probability ϕ_H that H holds with a Beta distribution, and the prior over $\boldsymbol{\phi} = (\phi_H)_{H \in \mathcal{H}_k}$ factorizes as

$$p_{k-1}(\boldsymbol{\phi}) = \prod_{H \in \mathcal{H}_k} \mathrm{Beta}\big(\phi_H;\, \alpha_{k-1}(H),\, \beta_{k-1}(H)\big), \tag{1}$$

where $\alpha_{k-1}(H)$ and $\beta_{k-1}(H)$ summarize the evidence from the first $k-1$ trajectories.

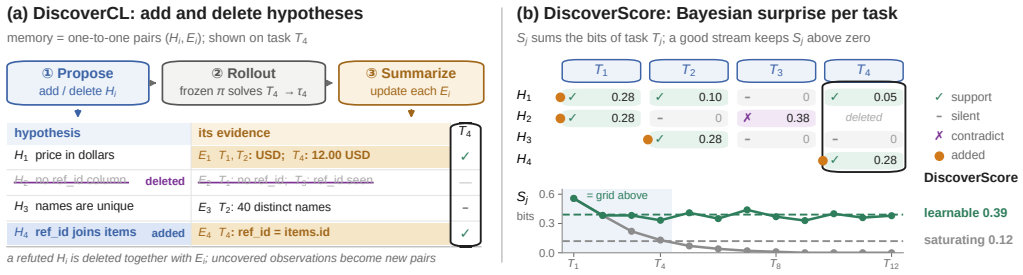

Figure 2: **Overview of DISCOVERCL and DISCOVERSCORE.** (**a**) DISCOVERCL: the memory pairs each hypothesis with its evidence; the proposer only adds and deletes hypotheses, and the summarizer only updates evidence by judging each hypothesis supported, contradicted, or silent. (**b**) DISCOVERSCORE: each judgment yields a per-hypothesis surprise $S_k(H)$ in bits; summing over the active hypotheses gives the per-task score S_k, which stays above zero on a learnable stream but vanishes on a saturating one.

Bayesian surprise. In a trajectory τ_k, evidence stems strictly from observations rather than actions. The key observation is that a task's educational value can be rigorously quantified—without needing external ground truth—simply by measuring how much these observations shift the agent's prior beliefs. Let $s_k(H)$ and $r_k(H)$ count the observations in τ_k that support and contradict a hypothesis $H \in \mathcal{H}_k$. By conjugacy, the posterior of ϕ_H is $\mathrm{Beta}\big(\alpha_k(H), \beta_k(H)\big)$ with

$$\alpha_k(H) = \alpha_{k-1}(H) + s_k(H), \qquad \beta_k(H) = \beta_{k-1}(H) + r_k(H). \tag{2}$$

The Bayesian surprise (Itti & Baldi, 2005) of task T_k on H is the Kullback–Leibler divergence between the posterior and the prior of ϕ_H, measured in bits:

$$S_k(H) = D_{\mathrm{KL}}\big(\mathrm{Beta}(\alpha_k(H), \beta_k(H)) \,\big\|\, \mathrm{Beta}(\alpha_{k-1}(H), \beta_{k-1}(H))\big). \tag{3}$$

Silence ($s_k(H) = r_k(H) = 0$) yields $S_k(H) = 0$, whereas any supporting or refuting evidence yields $S_k(H) > 0$. Because beliefs factorize across the active set $\mathcal{H}_k$, which is fixed before T_k begins, the surprise of the whole task is simply the sum of $S_k(H)$ over $\mathcal{H}_k$.

Maintaining the hypothesis set. While Bayes' rule updates beliefs within a fixed set, an open-ended stream requires the hypothesis set itself, and hence the dimension of $\boldsymbol{\phi}$, to evolve. After task T_k, we therefore apply two additional operations: **(1)** Addition: novel observations in τ_k that fall outside the coverage of $\mathcal{H}_k$ generate new, independent hypotheses, forming a new cohort $\mathcal{N}_{k+1}$ initialized with a $\mathrm{Beta}(1,1)$ prior, i.e., $\alpha_k(H) = \beta_k(H) = 1$. **(2)** Deletion: because a hypothesis acts as a universal claim, a single contradiction is fatal, and any hypothesis H conflicting with new evidence ($r_k(H) > 0$) is permanently removed. The next hypothesis set is thus

$$\mathcal{H}_{k+1} = \{H \in \mathcal{H}_k : r_k(H) = 0\} \cup \mathcal{N}_{k+1}. \tag{4}$$

Surviving hypotheses progress to $\mathcal{H}_{k+1}$ verbatim, retaining their exact cumulative support counts $\alpha_k(H) - 1$ (and $\beta_k(H) = 1$, as they have never been contradicted). Under this idealization, an unrefuted hypothesis cannot be silently forgotten: it survives with its accumulated support, while a refuted hypothesis is removed explicitly (Agarwal et al., 2025).

5 DESIGNING A META-HARNESS FOR AGENTIC CL

Section 4 defined an ideal Bayesian learner that maintains a hypothesis set to measure knowledge discovery. We now translate this theoretical framework into a versatile meta-harness, DISCOVERCL. This harness wraps around any frozen LLM agent, translating abstract beliefs into textual formats that naturally exploit the in-context learning (ICL) capabilities of LLMs for continual learning.

Memory. The memory stores the hypothesis set as a list $\mathcal{H}_k = (H_1, \ldots, H_{m_k})$ paired with textual evidence $\mathcal{E}_k = (E_1, \ldots, E_{m_k}, E_\varnothing)$, forming the overall state $\mathcal{M}_k = (\mathcal{H}_k, \mathcal{E}_k)$. Each E_j records the observations bearing on H_j up to task T_k, serving as a readable counterpart of the Beta parameters $\big(\alpha_k(H_j), \beta_k(H_j)\big)$, while $E_\varnothing$ collects observations not covered by any hypothesis (Step 3). The motivation for this design is rooted in LLMs' strength in ICL: instead of struggling to maintain

Algorithm 1 DISCOVERCL meta-harness for agentic CL

Require: Stream $(T_1, \ldots, T_K)$, frozen policy π, $\mathrm{LLM}_{\mathrm{prop}}$, $\mathrm{LLM}_{\mathrm{sum}}$; $\mathcal{H}_0 \leftarrow ()$, $\mathcal{E}_0 \leftarrow ()$

1: **for** $k = 1, \ldots, K$ **do**
2: $\quad \mathcal{H}_k \leftarrow \mathrm{LLM}_{\mathrm{prop}}(\mathcal{H}_{k-1}, \mathcal{E}_{k-1})$ ▷ Step 1: Propose, Eq. 5
3: $\quad \tau_k \sim \pi(\cdot \mid T_k, \mathcal{H}_k, \mathcal{E}_{k-1})$ ▷ Step 2: Roll out, Eq. 6
4: $\quad \mathcal{E}_k \leftarrow \mathrm{LLM}_{\mathrm{sum}}(\mathcal{E}_{k-1}, \mathcal{H}_k, \tau_k)$ ▷ Step 3: Summarize, Eq. 7
5: **end for**

abstract numerical parameters (like Beta distributions), the LLM can directly evaluate a hypothesis when it is explicitly conditioned on readable evidence. Hypotheses are sampled from the LLM based on past evidence, while the evidence itself is collected directly from observational data. To keep the hypotheses independent, as assumed in Equation 1, each is anchored to a distinct entity (e.g., the price column of table1).

Step 1: Propose hypotheses. Before task T_k begins, the agent must curate a clean, conflict-free set of beliefs that fits efficiently within the limited ICL context window. Thus, $\mathrm{LLM}_{\mathrm{prop}}$ refines the hypothesis set based on the past evidence $\mathcal{E}_{k-1}$ using the two strict operations of Section 4: **(1)** Addition, which turns the unmatched evidence in $E_\varnothing$ into new hypotheses, forming the cohort $\mathcal{N}_k$, and **(2)** Deletion, which permanently removes hypotheses whose evidence records a contradiction ($r_{k-1}(H) > 0$), preventing the LLM from hallucinating over refuted claims. Uncontradicted hypotheses are carried over verbatim, so this step realizes for the transition from $\mathcal{H}_{k-1}$ to $\mathcal{H}_k$:

$$\mathcal{H}_k \leftarrow \mathrm{LLM}_{\mathrm{prop}}(\mathcal{H}_{k-1}, \mathcal{E}_{k-1}). \tag{5}$$

Step 2: Roll out the trajectory. The frozen policy π attempts to solve the task by taking the updated hypotheses $\mathcal{H}_k$ and the previous evidence $\mathcal{E}_{k-1}$ as its context. By explicitly providing the hypotheses together with their evidence, we empower the LLM's reasoning process: the hypotheses explicitly state what actionable rules the agent currently believes, while the evidence dictates the confidence to trust them. The stream provides T_k, but the policy's actions decide which hypotheses the trajectory actually tests.

$$\tau_k \sim \pi(\cdot \mid T_k, \mathcal{H}_k, \mathcal{E}_{k-1}) \tag{6}$$

Step 3: Summarize evidence. After the task concludes, the agent must extract knowledge while making belief revision auditable. We therefore isolate evidence collection: $\mathrm{LLM}_{\mathrm{sum}}$ judges each hypothesis $H \in \mathcal{H}_k$ as supported, contradicted, or silent from the environment observations in τ_k and updates *only* the evidence. The agent's actions and task score are not treated as evidence. Two cases require special handling: first, observations about entities lacking a corresponding hypothesis are stored in the unmatched evidence bin $E_\varnothing$, which fuels future additions; second, a contradiction is explicitly recorded in the evidence of the contradicted hypothesis, marking it for deletion in the subsequent Step 1.

$$\mathcal{E}_k \leftarrow \mathrm{LLM}_{\mathrm{sum}}(\mathcal{E}_{k-1}, \mathcal{H}_k, \tau_k) \tag{7}$$

Overall. Ultimately, this harness operationalizes the Bayesian discovery loop of Section 4. The main difficulty in agentic CL is continuously maintaining a broad enough hypothesis set to ensure the uncontrolled task stream consistently generates effective learning signals. By splitting the updates—$\mathrm{LLM}_{\mathrm{prop}}$ writes only $\mathcal{H}$ while $\mathrm{LLM}_{\mathrm{sum}}$ writes only $\mathcal{E}$—the loop approximates the verbatim survival described in Section 4 and ties each hypothesis change to recorded evidence. Step 3 also supplies the counts $s_k(H), r_k(H)$ of Equation 3, extracting the evidence needed to compute Bayesian surprise without another model call. Overall, the harness adds two LLM inferences per task.

6 ANALYZING TASK STREAMS FOR AGENTIC CL

An agentic CL benchmark score often confounds two causes: (1) the agent may learn poorly, or (2) the stream itself may offer little to learn. In this section, we show how we can estimate this surprise empirically from the judgments that DISCOVERCL already records.

Estimating Bayesian surprise. For each task in the task stream, DISCOVERCL returns one judgment per hypothesis and task, so we binarize the evidence counts: $s_k(H), r_k(H) \in \{0, 1\}$, with at most one of them non-zero. Because a contradicted hypothesis is deleted, every active hypothesis $H \in \mathcal{H}_k$ enters T_k with a prior of $\mathrm{Beta}(n + 1, 1)$, where $n = \alpha_{k-1}(H) - 1$ counts the earlier

Table 1: **Evaluation results on DISCOVERCL with CL-BENCH.** Streams evaluated (left to right): Cohort Studies, Database Exploration (Database), Blind Spectrum Monitoring (Spectrum), Exploitable Poker, Sales Prediction, and Codebase Adaptation. **Score** denotes the mean reward ± standard deviation across runs (higher is better). **#Mem** indicates the average memory footprint in thousands of characters, which varies significantly due to differing agent trajectories. **Full ICL** refers to full-context in-context learning. For each backbone, the darkest and second-darkest gray shading highlight the best and second-best scores per column, respectively.

	Cohort		Database		Spectrum		Poker		Sales		Codebase	
Method	Score	#Mem	Score	#Mem	Score	#Mem	Score	#Mem	Score	#Mem	Score	#Mem
Claude Sonnet 5												
Base	$-.001_{\pm.00}$	0	$.179_{\pm.02}$	0	$.220_{\pm.00}$	0	$.426_{\pm.30}$	0	$.788_{\pm.01}$	0	$.297_{\pm.03}$	0
Full ICL	$.032_{\pm.01}$	89.0	$.488_{\pm.06}$	47.0	$.384_{\pm.04}$	65.6	$2.027_{\pm.33}$	173.1	$.788_{\pm.02}$	91.3	$.468_{\pm.02}$	212.4
AWM	$-.012_{\pm.01}$	3.2	$.156_{\pm.06}$	2.6	$.220_{\pm.00}$	1.9	$.620_{\pm.11}$	3.5	$.749_{\pm.07}$	4.1	$.338_{\pm.02}$	3.3
MemP	$-.005_{\pm.01}$	8.1	$.267_{\pm.07}$	3.1	$.284_{\pm.01}$	4.7	$.732_{\pm.16}$	5.6	$.662_{\pm.00}$	30.7	$.345_{\pm.06}$	14.6
ReasoningBank	$-.019_{\pm.00}$	3.5	$.162_{\pm.02}$	2.4	$.220_{\pm.00}$	2.3	$.608_{\pm.40}$	2.2	$.756_{\pm.04}$	3.4	$.385_{\pm.01}$	2.8
DISCOVERCL	$-.001_{\pm.01}$	6.4	$.388_{\pm.12}$	11.9	$.415_{\pm.02}$	9.6	$.473_{\pm.25}$	11.0	$.834_{\pm.02}$	4.7	$.389_{\pm.01}$	6.9
Claude Opus 5												
Base	$-.006_{\pm.00}$	0	$.358_{\pm.03}$	0	$.219_{\pm.00}$	0	$.675_{\pm.19}$	0	$.759_{\pm.04}$	0	$.642_{\pm.03}$	0
Full ICL	$-.004_{\pm.00}$	149.7	$.672_{\pm.02}$	42.2	$.568_{\pm.01}$	64.9	$1.290_{\pm.11}$	199.8	$.813_{\pm.00}$	70.9	$.820_{\pm.02}$	90.9
AWM	$-.005_{\pm.00}$	4.0	$.448_{\pm.03}$	6.1	$.256_{\pm.00}$	3.6	$.463_{\pm.15}$	6.9	$.769_{\pm.01}$	5.0	$.722_{\pm.00}$	3.9
MemP	$-.004_{\pm.00}$	15.2	$.333_{\pm.03}$	3.3	$.352_{\pm.00}$	4.6	$.590_{\pm.12}$	4.9	$.768_{\pm.00}$	35.3	$.726_{\pm.04}$	12.2
ReasoningBank	$.003_{\pm.00}$	3.5	$.391_{\pm.00}$	2.9	$.223_{\pm.00}$	2.7	$.049_{\pm.30}$	2.8	$.787_{\pm.00}$	3.3	$.774_{\pm.01}$	2.9
DISCOVERCL	$-.005_{\pm.00}$	8.6	$.661_{\pm.08}$	15.9	$.395_{\pm.09}$	14.7	$.833_{\pm.06}$	17.8	$.798_{\pm.02}$	7.4	$.741_{\pm.03}$	12.2

tasks that supported it ($n = 0$ for a newly added hypothesis). The per-hypothesis surprise $S_k(H)$ in Equation 3 then reduces to

$$S_k(H) = D_{\mathrm{KL}}\big(\mathrm{Beta}(n+1+s_k(H),\, 1+r_k(H)) \,\big\|\, \mathrm{Beta}(n+1,\, 1)\big). \tag{8}$$

Each judgment thus falls into one of three cases: (**1**) Silence where $s_k(H) = r_k(H) = 0$ leaves the prior unchanged and yields 0 bits; (**2**) Support where $s_k(H) = 1$ yields $D_{\mathrm{KL}}(\mathrm{Beta}(n+2,1) \,\|\, \mathrm{Beta}(n+1,1))$ bits, which diminish as the hypothesis becomes established: the first support carries 0.28 bits, while the fourth carries only 0.03 bits; (**3**) Refutation where $r_k(H) = 1$ yields $D_{\mathrm{KL}}(\mathrm{Beta}(n+1,2) \,\|\, \mathrm{Beta}(n+1,1))$ bits, which instead grow: 0.28 bits with no prior support, but 0.47 bits after three supports, whether H is then deleted or refined. This binarization allows the score to count tasks rather than raw observations.

Calculation of DISCOVERSCORE. DISCOVERSCORE (DS) quantifies the amount of information (in bits) that task T_k brings, strictly conditioned on the sequence of previous tasks. The history $T_1, \ldots, T_{k-1}$ shapes this score through the active set $\mathcal{H}_k$ and its support counts. The score of T_k is the total Bayesian surprise it causes on $\mathcal{H}_k$ (with each term given by Equation 8), and the overall score for the entire stream is the average per-task surprise:

$$\mathrm{DS}(T_k \mid T_{1:k-1}) = \sum_{H \in \mathcal{H}_k} S_k(H), \quad \mathrm{DS}(T_{1:K}) = \frac{1}{K}\sum_{k=1}^{K} \mathrm{DS}(T_k \mid T_{1:k-1}). \tag{9}$$

Because the agent's policy decides what each trajectory observes (Step 2), DISCOVERSCORE characterizes a coupled model–stream pair rather than the stream in isolation.

Evaluating agentic CL benchmarks. Crucially, a valid evaluation environment for agentic CL requires that the information measured by DISCOVERSCORE does not rapidly decay. A stream that saturates stops producing novel observations; consequently, births of new hypotheses cease and, since repeated support carries ever less surprise, S_k decays toward zero. Therefore, a robust agentic CL benchmark must provide a continuously evolving stream where later tasks consistently offer new evidence to either test existing hypotheses or spawn new ones.

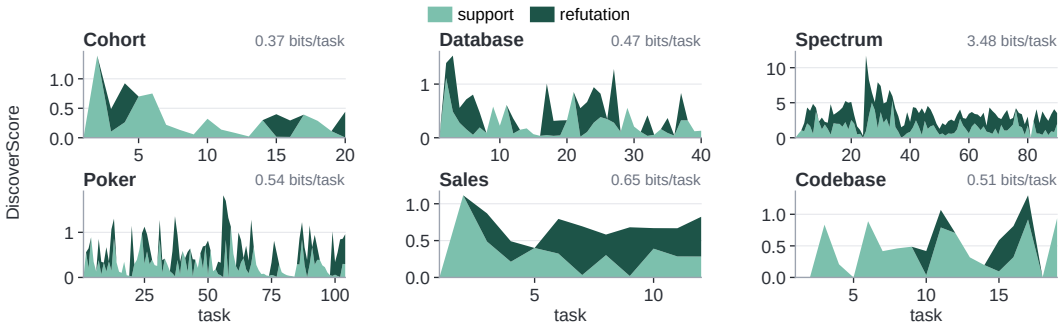

Figure 3: **DISCOVERSCORE with Claude Sonnet 5 on CL-BENCH.** Each panel shows the running mean through task k, decomposed into contributions from supporting (light) and refuting (dark) evidence. The value in the upper-right corner is the full-stream average in bits per task (Equation 9); Poker uses the 104 completed hands. Each panel has its own y-scale.

7 EXPERIMENTAL SETTINGS

Evaluation. We evaluate on the six streams of CL-BENCH (Asawa et al., 2026), whose instances share latent structure that can be discovered online: **(1)** Cohort Studies, patient cohorts with shared survival factors; **(2)** Database Exploration, an SQLite database with obfuscated schema conventions; **(3)** Blind Spectrum Monitoring, a radio band with hidden channels; **(4)** Exploitable Poker, heads-up poker against exploitable opponents; **(5)** Sales Prediction, store-level demand forecasting; and **(6)** Codebase Adaptation, bug fixing in recurring repositories. Each stream keeps its original task order and metric, and memory is reset between streams. The evaluated traces contain all default instances except Poker, for which 104 of the planned 120 hands completed.

Baselines and models. We use Claude Sonnet 5 and Claude Opus 5 as backbones, with the same backbone and interaction budget for all methods. Baselines are **Base** (no memory), **Full-context ICL** (raw preceding trajectories in context), and three memory methods: AWM (Wang et al., 2025), which induces reusable workflows from past successful trajectories and adds them to the agent's context; MEMP (Fang et al., 2025b), which distills trajectories into procedural memory, pairing step-level instructions with higher-level scripts, and updates it as new tasks arrive; and REASONINGBANK (Ouyang et al., 2026), which extracts generalizable reasoning strategies from both self-judged successes and failures and retrieves the relevant ones for each new task. DISCOVERSCORE and all analysis experiments use Claude Sonnet 5.

8 EXPERIMENTAL RESULTS

DISCOVERCL can adapt to different types of tasks. As a meta-harness, DISCOVERCL applies unchanged to all six streams, from database exploration to poker and codebase adaptation. It matches or improves the base model in all 12 stream–backbone settings; across the 10 non-Cohort settings, the relative improvement averages 46%. Existing memories are less reliable: AWM, MemP, and ReasoningBank improve the base model by at most 21% on average, and each falls below it in 3 of these 10 settings. The largest gains come on Database and Spectrum (+80–117%), whose rewards depend on stable environment facts, precisely the declarative knowledge that falsifiable hypotheses capture.

Table 2: **Hypothesis-update operations (%).** Overall is weighted by verdict count.

Stream	Silence	Support	Refute
Cohort	32.3	57.9	9.7
Database	46.8	35.7	17.4
Spectrum	14.6	56.6	28.8
Poker	33.4	55.6	11.0
Sales	4.7	77.1	18.2
Codebase	16.1	80.4	3.6
Overall	**28.6**	**52.2**	**19.1**

DISCOVERCL remains competitive with Full ICL using 10% of its memory. Full ICL keeps every past trajectory in context and provides a high-context reference, but not a strict upper bound. Across all 12 settings, DISCOVERCL uses 9.8% as much memory on average (10.6K vs. 108.1K characters). With Claude Opus 5 on Database, Spectrum, Sales, and Codebase, it recovers 50–97% of Full ICL's gain over Base while using 2.6–9.6× less memory. Its memory is larger than ReasoningBank's 2–4K characters but adapts to the stream. DISCOVERCL even surpasses Full ICL

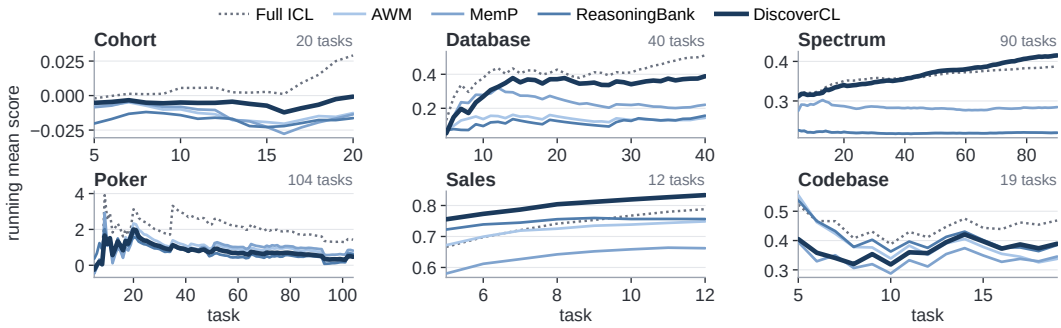

Figure 4: **Online reward with Claude Sonnet 5 on CL-BENCH.** Each line shows the running mean reward through task k (higher is better), drawn from task 5 onward because the earliest means are dominated by individual tasks. Full ICL (dotted) retains all preceding trajectories; each panel has its own y-scale. The Poker curves contain 104 completed hands from the 120-hand schedule.

on Spectrum and Sales with Claude Sonnet 5, suggesting that tested hypotheses filter noise from raw history. The main exception is Sonnet 5 on Poker (0.473 vs. 2.027): exploiting a stochastic opponent requires frequency estimates that raw hands retain but universal claims, retired at the first contradiction, do not.

DISCOVERSCORE diagnoses how strongly a task stream exercises continual learning. DISCOVERSCORE distinguishes a genuinely continual-learning stream from a mere sequence of tasks: a high, sustained score means that later tasks repeatedly test and revise beliefs formed earlier, whereas a low score indicates that the stream provides little cross-task evidence. Spectrum most faithfully exhibits the former structure. Each scan tests hypotheses about hidden channels, 28.8% of its verdicts refute an existing hypothesis (Table 2), and these corrections remain important throughout the stream. It consequently has the highest DISCOVERSCORE, at 3.48 bits per task (Figure 3). Cohort, by contrast, provides only 0.37 bits per task. Thus, DISCOVERSCORE can diagnose whether a benchmark merely orders tasks sequentially or actually requires an agent to learn continually by correcting and carrying forward its beliefs.

9 DISCUSSION

Q1: What does DISCOVERCL store in memory? The memory stores a variable-size set of hypotheses paired with the evidence accumulated for each one. Table 3 reports the Claude Sonnet 5 memory in characters and shows that the representation adapts its granularity to the stream. Spectrum maintains many short pairs (21.8 on average), with 210 hypothesis characters and 222 evidence characters per pair, reflecting the many channels tracked independently. For example, one hypothesis states that the transmitter at 79.8 MHz persists across scans with a bandwidth of about 15 MHz, while its evidence aggregates observation counts and the measured frequency, power, and bandwidth ranges. Database instead maintains only 6.2 pairs, but each hypothesis averages $1{,}658$ characters because it must describe a richer schema convention. The other streams retain 4.1–8.1 pairs.

Table 3: **Memory statistics.** Average pair counts and characters per pair; gray marks the top two per column.

Stream	#Pairs	Hyp. chars/pair	Evid. chars/pair
Cohort	6.8	658	459
Database	6.2	1,658	545
Spectrum	21.8	210	222
Poker	8.1	647	451
Sales	4.1	476	455
Codebase	7.5	589	441

Q2: Does DISCOVERCL induce stronger continual learning than other memories? The Sonnet 5 reward curves in Figure 4 show how each method converts accumulated experience into later performance. On Database, Spectrum, Sales, and Codebase, DISCOVERCL finishes above every other bounded-memory method, and its advantage generally emerges or persists as more tasks arrive. This sustained separation indicates that its hypothesis–evidence memory converts sequential observations into reusable knowledge more reliably than workflow and procedural memories. Poker is the main exception: exploiting a stochastic opponent requires frequency estimates that raw histories and procedural memories retain better than universal hypotheses. Thus, when a stream repeatedly tests falsifiable environment regularities, DISCOVERCL induces stronger continual improvement than the other bounded-memory baselines.

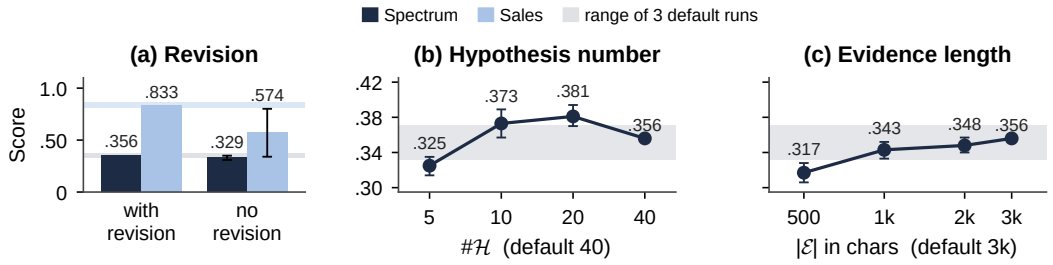

Figure 5: **Ablation studies on DISCOVERCL.** Scores use the first 40 Spectrum tasks and all 12 Sales tasks. (**a**) With and without hypothesis revision. (**b**) Number of active hypotheses on Spectrum. (**c**) Evidence length on Spectrum. Gray bands show the range of three default runs.

10 ABLATION STUDIES

Table 1 analyzes the model backbone on the complete evaluated streams. Figure 5 uses separate Claude Sonnet 5 ablation runs: the first 40 Spectrum tasks and all 12 Sales tasks. Its default Spectrum score (.356) therefore differs from the 90-task full-stream score in Table 1 (.415). Scores are reported directly; whiskers show 95% bootstrap intervals, and gray bands show the range of three default runs on the same ablation subset.

Ablation on model backbones. As shown in Table 1, replacing Claude Sonnet 5 with Claude Opus 5 raises DISCOVERCL's reward on Database ($.388 \rightarrow .661$), Poker ($.473 \rightarrow .833$), and Codebase ($.389 \rightarrow .741$), while producing a larger hypothesis–evidence memory on every stream. Opus is not uniformly better on Cohort, Spectrum, or Sales, but the gains on the more reasoning-intensive streams suggest that backbone quality affects how completely the harness proposes hypotheses and summarizes evidence.

Ablation on hypothesis revision. Figure 5(a) shows that removing revision lowers the score from .356 to .329 on Spectrum and from .833 to .574 on Sales. Revision matters because both streams change over time: channel availability shifts across Spectrum stages, while Sales rotates across stores and years. Without revision, hypotheses formed under an earlier regime remain in context after they cease to describe the current one, so the policy acts on stale regularities. The larger Sales drop suggests that its shorter stages leave fewer subsequent tasks to expose and correct such errors.

Ablation on hypothesis number $|\mathcal{H}|$. Figure 5(b) shows a non-monotonic capacity effect on Spectrum, with $|\mathcal{H}| = 20$ giving the highest point estimate (.381), compared with .356 at the default 40, .373 at 10, and .325 at 5. Spectrum requires enough hypotheses to track multiple channels, which explains the degradation at 5. More capacity is not always better because the spectrum changes across stages and old hypotheses are removed only after a later scan explicitly contradicts them. A budget of 20 acts as useful regularization, retaining sufficient coverage while limiting interference from outdated beliefs.

Ablation on evidence length $|\mathcal{E}|$. Figure 5(c) follows the expected monotonic pattern: Spectrum improves from .317 at 500 characters to .343 at 1,000, .348 at 2,000, and .356 at the default 3,000. Blind Spectrum Monitoring requires the agent to accumulate evidence across scans—including observed frequency ranges, repeated availability, and contradictions to earlier channel hypotheses—rather than infer each answer from the current scan alone. Short evidence documents compress away precisely this history, weakening both hypothesis validation and later revision. Thus, unlike the hypothesis count, evidence capacity has no competing stale-belief effect and remains the tighter memory bottleneck.

11 CONCLUSION

We cast agentic continual learning as Bayesian discovery and introduced DISCOVERCL, which maintains evidence-grounded hypotheses, and DISCOVERSCORE, which measures belief change. DISCOVERCL is the strongest bounded-memory method in 9 of 12 settings. Future work could use these diagnostics to adapt memory updates, evidence collection, or task curricula automatically. This would turn stream diagnosis into a mechanism for improving learning. More broadly, discovery offers a common language for designing, diagnosing, and improving agentic CL.

ETHICS STATEMENT

Persistent textual memory may carry incorrect, stale, sensitive, or malicious observations into later actions, while erroneous summarizer verdicts can retain false hypotheses or discard valid ones. Our experiments are confined to existing CL-BENCH research environments and do not involve human subjects or real-world, safety-critical deployment; DISCOVERSCORE measures belief change rather than truth, reliability, or safety. Any deployment should minimize and segregate retained data, avoid personal or confidential content, sandbox tools, log and expose memory revisions, support expiry and deletion, and require human review before consequential actions. DISCOVERCL also adds two model calls per task, whose computational and environmental costs should be considered.

REPRODUCIBILITY STATEMENT

DISCOVERCL is training-free and adds proposer and summarizer calls around the frozen acting policy. Complete experimental details are provided in Appendix B, including the task streams, update order, prompts, policy context, hypothesis budget $B = 40$, target evidence budget $C = 3{,}000$ characters, and $3C$ hard cap. Section 7 defines the baselines and evaluation scope, while Equations 8–9 define the meta-evaluation. We preserve CL-BENCH's task order and metrics, reset memory between streams, and match backbones and interaction budgets. Reproductions should record exact model identifiers and access dates, decoding parameters and seeds when available, intermediate memories, and per-run scores; we report means, standard deviations, and paired 95% bootstrap intervals.

AI USE STATEMENT

Claude Sonnet 5 and Claude Opus 5 are the studied systems and experimental components, serving as frozen policies, proposers, and summarizers. Generative AI tools also assisted with language editing, manuscript organization, method presentation, and LaTeX preparation. The authors reviewed and revised all AI-assisted material and remain solely responsible for the paper's claims, citations, mathematics, methods, code, results, and conclusions.

REFERENCES

Dhruv Agarwal, Bodhisattwa Prasad Majumder, Reece Adamson, Megha Chakravorty, Satvika Reddy Gavireddy, Aditya Parashar, Harshit Surana, Bhavana Dalvi Mishra, Andrew McCallum, Ashish Sabharwal, and Peter Clark. AutoDiscovery: Open-ended scientific discovery via Bayesian surprise. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0847. URL https://papers.nips.cc/paper_files/paper/2025/hash/23b127521af7ca7a42f5cdb7507be4f2-Abstract-Conference.html.

Qingyao Ai, Yichen Tang, Changyue Wang, Jianming Long, Weihang Su, and Yiqun Liu. MemoryBench: A benchmark for memory and continual learning in LLM systems. *arXiv preprint arXiv:2510.17281*, 2025. URL https://arxiv.org/abs/2510.17281.

Parth Asawa, Christopher M. Glaze, Gabriel Orlanski, Ramya Ramakrishnan, Benji Xu, Asim Biswal, Vincent Sunn Chen, Frederic Sala, Matei Zaharia, and Joseph E. Gonzalez. Continual learning bench: Evaluating frontier ai systems in real-world stateful environments, 2026. URL https://arxiv.org/abs/2606.05661.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.05080.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022. doi: 10.1109/TPAMI.2021.3057446. URL https://www.computer.org/csdl/journal/tp/2022/07/09349197/1qYm8xg25JS.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*, 2025a. URL https://arxiv.org/abs/2510.18866.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring agent procedural memory. *arXiv preprint arXiv:2508.06433*, 2025b. URL https://arxiv.org/abs/2508.06433.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an AI co-scientist. *arXiv preprint arXiv:2502.18864*, 2025. URL https://arxiv.org/abs/2502.18864v1.

Qisheng Hu, Quanyu Long, and Wenya Wang. When continual learning moves to memory: A study of experience reuse in LLM agents. *arXiv preprint arXiv:2604.27003*, 2026. URL https://arxiv.org/abs/2604.27003.

Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems*, volume 18, 2005. URL https://proceedings.neurips.cc/paper/2005/hash/0172d289da48c48de8c5ebf3de9f7ee1-Abstract.html.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. DiscoveryWorld: A virtual environment for developing and evaluating automated scientific discovery agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0324. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/13836f251823945316ae067350a5c366-Abstract-Datasets_and_Benchmarks_Track.html.

Thomas Joshi, Shayan Chowdhury, and Fatih Uysal. SWE-Bench-CL: Continual learning for coding agents. *arXiv preprint arXiv:2507.00014*, 2025. URL https://arxiv.org/abs/2507.00014.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2308.03688.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024. URL https://arxiv.org/abs/2408.06292.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.25140.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 2019. doi: 10.1016/j.neunet.2019.01.012. URL https://arxiv.org/abs/1802.07569.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763. URL https://arxiv.org/abs/2304.03442.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi: 10.1038/s41586-023-06924-6. URL https://www.nature.com/articles/s41586-023-06924-6.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Yiheng Shu, Bernal Jiménez Gutiérrez, Saisri Padmaja Jonnalagedda, Yuguang Yao, Huan Sun, and Yu Su. AgentCL: Toward rigorous evaluation of continual learning in language agents. *arXiv preprint arXiv:2606.02461*, 2026. URL https://arxiv.org/abs/2606.02461.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.333. URL https://aclanthology.org/2026.eacl-long.333/.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024a. URL https://openreview.net/forum?id=ehfRiF0R3a.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024b. doi: 10.1109/TPAMI.2024.3367329. URL https://www.computer.org/csdl/journal/tp/2024/08/10444954/1URb3WOOpNK.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.775. URL https://aclanthology.org/2022.emnlp-main.775/.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25bx.html.

Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H. Chi, Chi Wang, Shuo Chen, Fernando Pereira, Wang-Cheng Kang, and Derek Zhiyuan Cheng. Evo-Memory: Benchmarking LLM agent test-time learning with self-evolving memory. *arXiv preprint arXiv:2511.20857*, 2025. URL https://arxiv.org/abs/2511.20857.

Cheng-Kuang Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Chen, and Hung-yi Lee. StreamBench: Towards benchmarking continuous improvement of language agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3398. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c189915371c4474fe9789be3728113fc-Abstract-Datasets_and_Benchmarks_Track.html.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. AgentGym: Evolving large language model-based agents across diverse environments. *arXiv preprint arXiv:2406.04151*, 2024. URL https://arxiv.org/abs/2406.04151.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1650. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Weijia Xu, Alessandro Sordoni, Chandan Singh, Zelalem Gero, Michel Galley, Xingdi Yuan, and Jianfeng Gao. Test-time learning with an evolving library. *arXiv preprint arXiv:2605.14477*, 2026. URL https://arxiv.org/abs/2605.14477.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025. URL https://arxiv.org/abs/2504.08066.

Xidong Yang, Xingyi Zhang, Wenhao Li, Wenyan Liu, Junjie Sheng, Yun Hua, Wei Yin, Tao Fang, Chuyun Shen, and Xiangfeng Wang. PATH-Bench: Path-dependent evaluation of lifelong agents. *arXiv preprint arXiv:2608.01149*, 2026. URL https://arxiv.org/abs/2608.01149.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022. URL https://arxiv.org/abs/2210.03629.

Mert Yuksekgonul, Daniel Koceja, Xinhao Li, Federico Bianchi, Jed McCaleb, Xiaolong Wang, Jan Kautz, Yejin Choi, James Zou, Carlos Guestrin, and Yu Sun. Learning to discover at test time. *arXiv preprint arXiv:2601.16175*, 2026. URL https://arxiv.org/abs/2601.16175.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.04618.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936. URL https://ojs.aaai.org/index.php/AAAI/article/view/29936.

Junhao Zheng, Xidi Cai, Qiuke Li, Duzhen Zhang, ZhongZhi Li, Yingying Zhang, Le Song, and Qianli Ma. LifelongAgentBench: Evaluating LLM agents as lifelong learners. *arXiv preprint arXiv:2505.11942*, 2025. URL https://arxiv.org/abs/2505.11942.

Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Duzhen Zhang, Chenxing Li, Dong Yu, and Qianli Ma. Lifelong learning of large language model based agents: A roadmap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. URL https://www.computer.org/csdl/journal/tp/2026/05/11328884/2d28VQTsIBa.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.13854.

A LIMITATIONS

A.1 METHOD LIMITATIONS

DISCOVERCL depends on LLM-based proposal and summarization, so correlated errors may preserve an incorrect hypothesis rather than refute it. The hypothesis and evidence budgets bound the added context, but a fixed evidence budget forces lossy compression on long streams, while verbatim write-back lets a hypothesis that no task contradicts persist indefinitely even if it is too vague to be useful. Our estimate also treats hypotheses independently and a single marked contradiction as decisive, so a false refutation by the summarizer is counted as learning.

A.2 EVALUATION SCOPE

DISCOVERSCORE depends on the policy, model, prompts, summarizer accuracy, and preceding task history rather than being an intrinsic property of a benchmark, and we report it only with Claude Sonnet 5. We preserve the prescribed order of each CL-BENCH stream and do not compare task permutations; our claims therefore diagnose the learning structure of the evaluated sequence, not robustness to alternative curricula. Our reward evaluation covers two models and six streams and does not establish robustness under longer horizons, distribution shift, or safety-critical deployment.

B EXPERIMENTAL DETAILS

B.1 BENCHMARK AND EVALUATION PROTOCOL

The six default CL-BENCH schedules (Asawa et al., 2026) contain 301 planned task instances. Our evaluated traces contain 285 instances because Poker completed 104 of its planned 120 hands; all other streams are complete. The reported Poker means and curves use these 104 completed hands, without imputing the incomplete instances. Table 4 records the evaluated lengths, stage structure, interaction type, and task reward. Instances follow the benchmark order; persistent memory is retained within a stream and reset between streams; and the original observations, feedback channels, and evaluators are unchanged.

Table 4: **CL-BENCH streams used in our evaluation.** K_{eval} is the number of evaluated task instances.

Stream	K_{eval}	Schedule	Interaction and reward
Cohort Studies	20	Five studies, four instances each	Up to 20 database-tool actions; information gain in survival prediction over a study-wide baseline, measured in bits per cohort.
Database Exploration	40	20 questions before and 20 after schema migration	Exploratory SQL queries followed by an answer; normalized regret rewards correct answers using fewer queries.
Blind Spectrum Monitoring	90	Three stages of 30 scans over a 168 MHz band	One scan report per instance; intersection-over-union of predicted and true long-run available spectrum.
Exploitable Poker	104	104 completed hands from the planned $20/30/10/35/25$ stages	One complete heads-up hold'em hand; average profit in big blinds per hand.
Sales Prediction	12	Three stages of four years, rotating three stores	Multi-step analysis in a persistent workspace; WAPE skill over 75 product–location–year forecasts.
Codebase Adaptation	19	Nine `tablib` issues, then ten `tenacity` issues	Containerized code editing; test-verified success weighted by the number of commands used.

B.2 Model access and licenses

Claude Sonnet 5 and Claude Opus 5 were accessed through Anthropic's hosted API; their use is governed by the Anthropic Commercial Terms of Service and Usage Policy, and no model weights are redistributed. The CL-BENCH code, configurations, and bundled benchmark-generated task assets are released under the Apache License 2.0, except where an upstream component retains its own license. Codebase Adaptation evaluates the upstream `tablib` and `tenacity` repositories, licensed under MIT and Apache 2.0, respectively, and uses the MIT-licensed SWE-bench and mini-swe-agent tooling. We retain upstream attribution and do not relicense third-party content.

B.3 Prompt templates

The following instructions instantiate the proposer and summarizer in Algorithm 1.

B.3.1 Proposer

1. Re-derive from the evidence what counts as an entity for new hypotheses and write it as a hypothesis template; existing unmarked hypotheses keep their names and text.
2. Write one hypothesis per entity, subject to the hypothesis budget B.
3. Before adding a new hypothesis, merge it into an existing one if both name the same thing or one names a property of the other; merging never rewrites an existing unmarked hypothesis.
4. State regularities as hypotheses and keep measurements in the evidence document.
5. State which observation would refute each hypothesis.
6. Omit a hypothesis only if its entry is marked `CONTRADICTED`, and write in its position a hypothesis supported by the new observations.
7. Copy every unmarked hypothesis verbatim; do not reword it, merge it for brevity, or drop it because the latest task did not mention it.
8. Read the uncovered queue first and decide which observations to promote to hypotheses.

B.3.2 Summarizer

1. Write one entry per hypothesis, in hypothesis order, under the hypothesis's exact name.
2. Store all facts about its entity as counts and ranges, not as a list of episodes.
3. Declare contradictions first: if the new observations cannot hold together with the hypothesis, begin the entry with `CONTRADICTED`; lack of support is not a contradiction.
4. Carry forward the existing content of entries whose entity is absent from the new observations.
5. Place observations that no hypothesis names only in the uncovered queue.

The summarizer receives τ_k but is instructed to use only its environment observations as evidence, excluding the agent's messages, actions, and task score.

B.4 Memory and context constraints

We use at most $B = 40$ active hypotheses and a target evidence budget of $C = 3{,}000$ characters. Existing unmarked hypotheses have priority when the hypothesis budget limits new admissions, and the evidence output has a hard cap of $3C$ to prevent runaway generations. For DISCOVERCL, no raw trajectories are placed in context, so $(\mathcal{H}_k, \mathcal{E}_{k-1})$ is the only cross-task information available to the policy.